SNIA CSTI | CLOUD STORAGE TECHNOLOGIES

Kubernetes in the Cloud – Part One

Live Webcast
May 2, 2019
11:00 am PT

Today's Presenters

Matt Baldwin
NetApp

Ingo Fuchs
NetApp

Mike Jochimsen
Kaminario

SNIA Legal Notice

SNIA CSTI | CLOUD STORAGE TECHNOLOGIES

- This work, "Kubernetes in the Cloud", is a derivative of "Intro to Kubernetes - from containers to orchestrating the world" by Alex Sirbu, used under CC BY. "Kubernetes in the Cloud" is licensed under CC BY by SNIA.
- This presentation is a project of the SNIA.
- Neither the author nor the presenter is an attorney and nothing in this presentation is intended to be, or should be construed as legal advice or an opinion of counsel. If you need legal advice or a legal opinion please contact your attorney.
- The information presented herein represents the author's personal opinion and current understanding of the relevant issues involved. The author, the presenter, and the SNIA do not assume any responsibility or liability for damages arising out of any reliance on or use of this information.

NO WARRANTIES, EXPRESS OR IMPLIED. USE AT YOUR OWN RISK.

SNIA-At-A-Glance

185
industry leading organizations

2,000
active contributing members

50,000
IT end users & storage pros worldwide

SNIA CSTI | CLOUD STORAGE TECHNOLOGIES

What We Do

Educate vendors and users on cloud storage, data services and orchestration

Support & promote business models and architectures: OpenStack, Software Defined Storage, Kubernetes, Object Storage

Understand Hyperscaler requirements Incorporate them into standards and programs

Collaborate with other industry associations

Kubernetes in the Cloud

Contents

SNIA CSTI | CLOUD STORAGE TECHNOLOGIES

- Containers
- Container Orchestrators
- Kubernetes - What? Why? How?
- Quick demo
- Q&A

Container - What's in a Name?

- Coming from the shipping industry
- Caused aquatic theme for domain

Shipping Containers

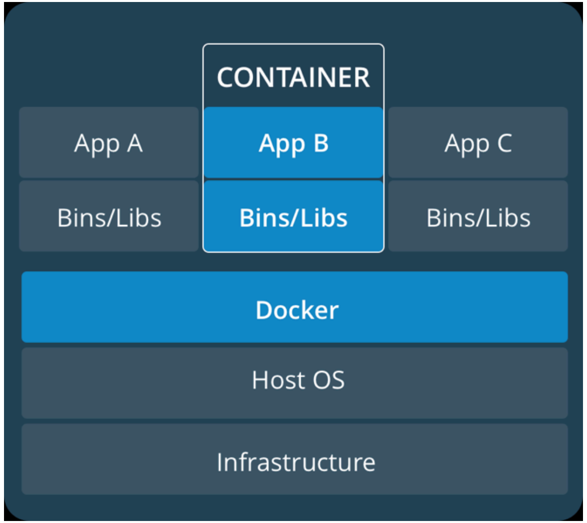
- Portability - can be used on any of supported types of ships
- Wide variety of cargo that can be packed inside
- Standard sizes - standard fittings on ships
- Many containers on a ship
- Isolates cargo from each other

Translated to Software

- Portability - can be used on any supported system (system with container execution environment)
- Wide variety of software that can be packed inside
- Standard format
- Many containers to a physical node
- Isolates execution of one container from another

What is a Container?

- Way to pack code and dependencies together
- Can run anywhere
- Execute multiple containers to a physical machine



Sounds Familiar?

- Same concept as virtual machines
- Pack OS and software together, to run in isolated instances
- Can run anywhere the specific hypervisor runs
- Multiple VMs to a physical machine

How do VMs Work?

- Hypervisor = layer between VM and kernel
- Emulates system calls
- Allows multiple types of operating systems on a machine (Windows on Linux)
- Overhead for hypervisor

Containers on the Other Hand ...

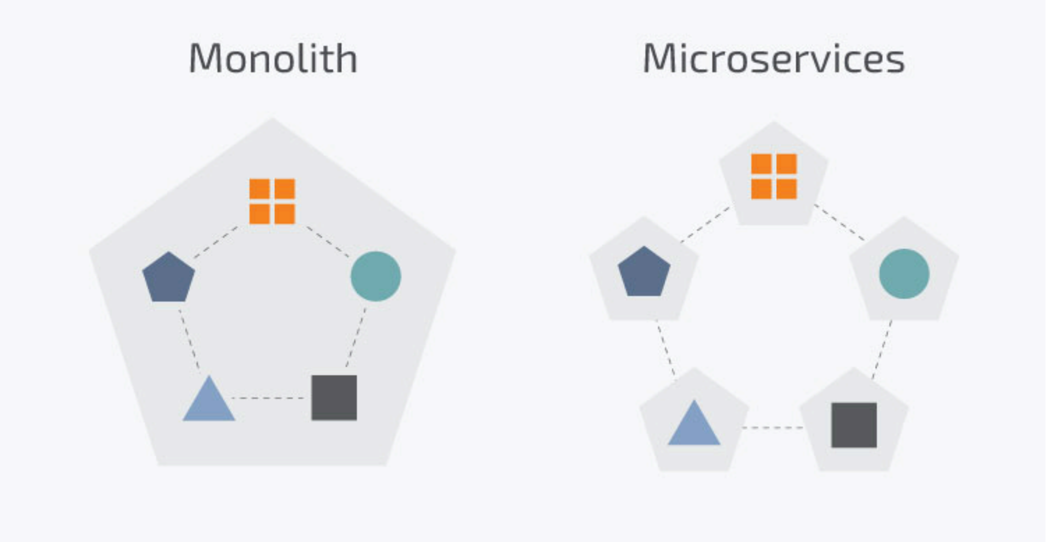
- Only contain application and application-related libraries and frameworks, that run on the host machine's kernel
- Smaller
- Lower overhead
- Differences in OS distributions and dependencies are abstracted - same kernel

Working Together, not against Each Other

- Windows on Linux possible only with VMs
- Older software needs to be adapted to be run as containers (and won’t)
- Usage of VMs as a medium for containers (better isolation and easier scaling)

Greater Modularity in Software

- Monolithic application → independent services that interact (microservices)



Containers Empowering Microservices

- Quicker start times -> easy to prototype or scale
- Allow work to be done independently on modules -> independent releases for components (take care of interfaces)
- Isolated and abstracted runtime environments, that can be tailored for each module
- Shared runtime environment, for heterogenous applications

Containers History – Early Days

- Need for resources to be shared among many users -> multiple terminals connected to the same mainframe
- Main problem - execution can cause the main computer to crash -> down for everybody

Containers History – Isolating More and More

- Chroot – 1979 – change root directory for a running process, along with children → segregate and isolate processes, protecting global environment
- Jails – additional process sandboxing features for isolating filesystems, users, networks (limiting apps in their functionality)
- Solaris Zones – full application environments, with full user, process and filesystem space
- Cgroups – 2006 – process containers designed for isolating and limiting the resource usage of a process

Containers History – Linux Containers (lxc)

2008

- Provides virtualization at OS level
- Provides containers with its own process and network space

Containers History – Docker

2013

- Container execution and management system
  - Originally started with lxc, then moved to libcontainer, which allows containers to work with:
  - linux namespaces
  - libcontainer control groups
  - capabilities
  - app armor security profiles
  - network interfaces
  - firewall rules

Containers History – OCI & CNCF

- Open Container Initiative – 2015
  - Industry format for a container format and container runtime software for all platforms
  - Spend resources on developing additional software to support use of standard containers, instead of format alternatives
- Cloud Native Computing Foundation (CNCF) – 2015
  - Working on different projects to further standardize the Cloud Native open-source market:
    - Kubernetes, Etcd, Prometheus, Envoy, Harbor, and more
    - CSI, CNI, Containerd, and more

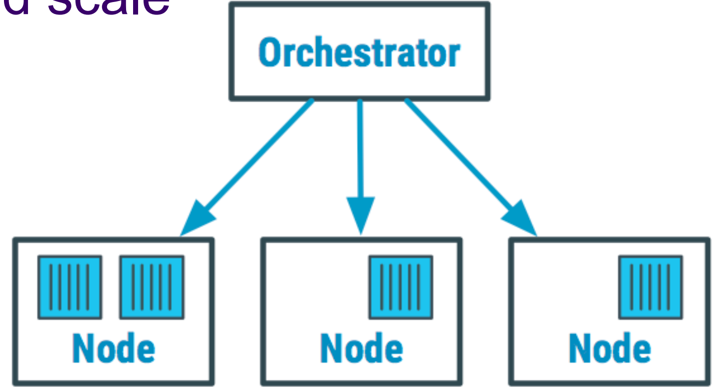
Need for Something More?

- Docker started out as a CLI on top of lxc, that built, created, started, stopped and exec'd containers
- Does management at a node level, upon specific requests
- Easy to manually manage with up to 100s of containers and 10s of nodes, but what next?

Container Orchestration

Orchestrator

- Manage and organize both hosts and docker containers running on a cluster
- Main issue - resource allocation - where can a container be scheduled, to fulfill its requirements (CPU/RAM/disk) + how to keep track of nodes and scale



Some Orchestrator Tasks

- Manage networking and access
- Track state of containers
- Scale services
- Do load balancing
- Relocation in case of unresponsive host
- Service discovery
- Attribute storage to containers
- ...

Kubernetes

What is Kubernetes?

- “Kubernetes” = Greek for governor, helmsman, captain
- Open-source container orchestration system
- Originally designed by Google, donated to and maintained by CNCF
- Aim to provide "platform for automating deployment, scaling and operations of application containers across clusters of hosts"

kubernetes

Why Kubernetes? - Goals

- Main objectives, stated by devs, for community
- Achieve velocity
- Allow scaling of both software and teams
- Present abstract infrastructure
- Gain efficiency

Achieve velocity

- Velocity = number of things you ship while maintaining a highly available service
- Achieved by:
  - Immutability - created artifact cannot be changed
  - Declarative configuration - declare desired state and Kubernetes' job is to ensure it matches
  - Self-healing systems - trying to maintain desired states if something changes

Allow scaling of software

- Encouraging decoupling in applications - separated components that communicate via defined APIs via load-balanced services
- Running in shared abstract environment, without interference
- Utilizing standard container format that runs on any machine

Allow Scaling of Teams

- Separation of concerns for consistency and scaling
- Application ops rely on the SLA provided by the platform
- Orchestrator ops uphold SLA

Present Abstract Infrastructure

- Decoupling container images and machines
- Cluster can be heterogenous and reduce overhead and cost
- Portability - container can be used on another cluster without being changed

Gain Efficiency

- Optimized usage of physical machines - multiple containers on same machine
- Isolated with namespaces, to not interfere with each other

Why Cloud?

- Use Kubernetes offered as managed service
- On demand use, due to quickness of setup of cluster + applications
- Automation, APIs, ...

DEMO

Questions

Join us Next Time…

Computing

Communication

Coordination

Application versioning

Stateful

After This Webcast

- Please rate this webcast and provide us with feedback
- This webcast and a PDF of the slides will be posted to the SNIA Cloud Storage Technologies Initiative website and available on-demand at https://www.snia.org/forum/csti/knowledge/webcasts
- A full Q&A from this webcast will be posted to the SNIA Cloud blog: www.sniacloud.com/
- Follow us on Twitter @SNIACloud

Thank You